

EXPRESS MAIL NO. EL652176636US #3IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Wenbo MAO)	Re: Claim to Priority
U.S. Appln. No.: not yet assigned)	Group: not yet assigned
U.S. Filing Date: concurrently herewith)	Examiner: not yet assigned
International Application No: PCT/GB00/00370)	
International Filing Date: 08 February 2000)	Our Ref.: B-4253PCT 618967-4
For: "VERIFICATION OF THE PRIVATE COMPONENTS OF A PUBLIC-KEY CRYPTOGRAPHIC SYSTEM")	Date: August 8, 2001

35 U.S.C. 119 CLAIM TO PRIORITY

Commissioner of Patents and Trademarks
 Box PCT
 Washington, D.C. 20231

Attn: United States Designated/Elected Office (DO/EO/US)

Sir:

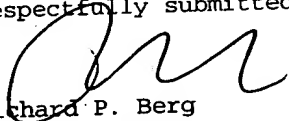
Prior PCT International Application No. PCT/GB00/00370,
 designating the U.S., claims foreign priority as follows:

<u>COUNTRY</u>	<u>FILING DATE</u>	<u>SERIAL NUMBER</u>
GB	08 February 1999	9902687.4

The certified copy has been filed in prior PCT International Patent Application No. PCT/GB00/00370.

Applicant hereby confirms that this claim for priority applies to the above-identified U.S. International stage application.

Respectfully submitted,


 Richard P. Berg
 Reg. No. 28,145
 Attorney for Applicant
 LADAS & PARRY
 5670 Wilshire Boulevard #2100
 Los Angeles, California 90036
 (323) 934-2300

THIS PAGE BLANK (USPTO)



The
Patent
Office

PCT/GB 00 / 00370
08 FEBRUARY 2000

INVESTOR IN PEOPLE

#3

GB00/370

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP9 1RH

REC'D 21 FEB 2000

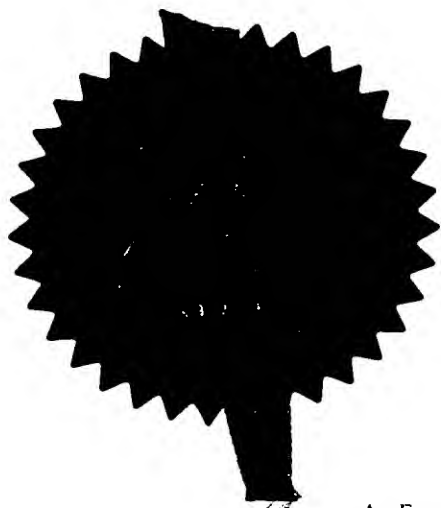
WIPO PCT

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.



Signed

G D Court

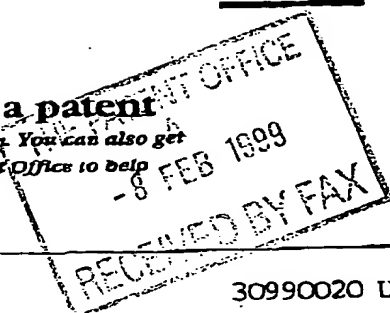
Dated

-1 APR 1999

THIS PAGE BLANK (USPTO)

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)



The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference

30990020 UK

2. Patent
(No)

9902687.4

8 FEB 1999

3. Full name, address and postcode of each applicant (underline all surnames)

HEWLETT-PACKARD COMPANY
3000 Hanover Street
Palo Alto
California 94304
United States of America

Patents ADP number (if you know it)

If the applicant is a corporate body, give the country/state of its incorporation

Delaware, USA

496588004

4. Title of the invention

Cryptographic Protocol

5. Name of your agent (if you have one)

Matthew John Mitchell Lawman
Hewlett-Packard Limited
IP Section
Filton Road
Stoke Gifford
Bristol BS34 8QZ

"Address for service" in the United Kingdom to which all correspondence should be sent (including the postcode)

Patents ADP number (if you know it)

7337009001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (if you know it) the or each application number

Country

Priority application number
(if you know it)

Date of filing
(day / month / year)

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application

Date of filing
(day / month / year)

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (Answer 'Yes' if:

YES

- a) any applicant named in part 3 is not an inventor, or
 - b) there is an inventor who is not named as an applicant, or
 - c) any named applicant is a corporate body.
- See note (d))

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form

Description 15

Claim(s) 1

Abstract

Drawing(s)

10. If you are also filing any of the following, state how many against each item.

Priority documents

Translations of priority documents

Statement of inventorship and right to grant of a patent (Patents Form 7/77)

Request for preliminary examination and search (Patents Form 9/77) 1

Request for substantive examination (Patents Form 10/77)

Any other documents 1 Fee Sheet
(please specify)

11.

I/We request the grant of a patent on the basis of this application.

Signature

Date 8/2/99

Matthew John Mitchell Lawman

12. Name and daytime telephone number of person to contact in the United Kingdom

Janet Smith 0117-922-8026

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

- a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.
- b) Write your answers in capital letters using black ink or you may type them.
- c) If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.
- d) If you have answered 'Yes' Patents Form 7/77 will need to be filed.
- e) Once you have filled in the form you must remember to sign and date it.
- f) For details of the fee and ways to pay please contact the Patent Office.

Patents Form 1/77

DUPLICATE

Toward Optimally Efficient Knowledge Demonstration of Solovay-Strassen Primality Evidence

Wenbo Mao
Hewlett-Packard Laboratories,
Filton Road, Stoke Gifford,
Bristol BS34 8QZ, United Kingdom.
wm@hplb.hpl.hp.com

February 8, 1999

Abstract

We construct the first example of an efficient proof of knowledge protocol for demonstrating Solovay-Strassen primality evidence on an integer being the product of two primes of roughly equal sizes. The computational cost of the new protocol is the same as that of performing six Solovay-Strassen primality tests on non-secret numbers of comparable sizes, with achieving the same error probability of 2^{-k} after k proof interactions. The new protocol also provides the first efficient proof of such integer structure without limiting to Blum integers.

1 Introduction

In public-key cryptography, an individual user's cryptographic key pair should be generated by the user himself. On the other hand, the user's public key should be certified by a known authority for authentication. The authority will naturally demand that a user-generated public/private key pair have a valid private component according to a set of agreed criteria. Zero-knowledge proof is a powerful tool that allows a user to run a protocol with an authority, convincing the latter of the structural validity of the private key without the former disclosing it. For instance, the ISO standardization document 9798 part 3 ([7]) recommends that public-key certification should include zero-knowledge proof for possession of the private component that matches the public key to be certified.

In this paper we propose a zero-knowledge protocol for proof of private-key structure for cryptosystems based on the difficulty of factoring (such as RSA). The structure is that an integer is the product of two primes of (roughly) equal size.

1.1 Background

A number of authors [1, 2, 4, 6, 8] proposed zero-knowledge, or proof-of-knowledge protocols for showing a general form of Blum integers $n = p^r q^s$ where p and q are two distinct primes satisfying $p \equiv q \equiv 3 \pmod{4}$, r and s are integers. These protocols use the following number theoretic observation. An integer n has two different prime factors, including their powers, if and only if a quarter of the elements in the multiplicative group modulo n are quadratic residues, and n will have at least three different prime factors if no more than one-eighth of such elements are quadratic residues. A prover can thus prove n in general form of Blum integer by demonstrating that for a set of randomly chosen elements as challenges, (s)he can display square roots modulo n for a quarter of them.

Boyar et al [3] proposed a zero-knowledge protocol for proof of square-free integers. An integer n is said to be square-free if there exists no factor p such that p^2 divides n . The working principle of that protocol is based on an observation that an integer n is square-free if and only if n is relatively prime to $\phi(n)$ (Euler's phi function), and this allows a prover to be able to consistently display an n -th root modulo n of a random challenge which is relatively prime to n .

If a general form of Blum integer is also square-free, then it must be the product of exactly two distinct primes (i.e., without containing their powers).

A desirable RSA modulus should be a square-free integer consisting of two prime factors of roughly equal sizes. Thus and obviously, proof of knowledge on that an integer is a desirable RSA modulus can go through applications of the previous techniques of proving a general form of Blum integer, proving square-free-ness, plus showing the sizes of the two factors (e.g., the approach of [5]). However, such an approach of applying several different proofs has two major disadvantages. First, it is inefficient. In particular we should notice that in square-root or n th-root displaying protocols, the proving/verification participants must agree on mutually trusted random challenges (or be served with such challenges by a mutually trusted random source). Otherwise, the prover can be used as an oracle to factor the integer under proof, and the verifier can be fooled easily. Therefore, in each of such protocols applied, some additional means, associated with additional costs, are necessary to let the two participants obtain mutually trusted random challenges (such costs were never discussed in the previous works (e.g., in [1, 2, 4, 6, 8, 5]). Secondly, for generality in applications and economy in choosing primes, RSA moduli should not be confined to Blum integers (even though some protocols require use of such integers). Efficient proof of knowledge on desirable RSA moduli

without limiting to Blum integers will also have a value in pursuit of new knowledge. In this paper we propose a proof of knowledge protocol to achieve exactly just that. The protocol shows the primality evidence using a variation of Solovay-Strassen primality test technique [9], at the same time shows the sizes of the factors almost at no additional cost. Compared with the previous approaches of applying several different proofs, the new protocol not only puts no constraint on the form of the prime factors (though they should be odd), but also reaches a new low in computational cost: the same as that of performing six Solovay-Strassen primality tests on non-secret integers of size of n . The error probability of the protocol is the same as that of Solovay-Strassen primality test: 2^{-k} after k interactions. Because Solovay-Strassen primality test is very efficient, and because there is no need to generate or agree on mutually trusted challenges, it is our belief that the new protocol reaches the best performance in comparison with the previous techniques for demonstration of primality in a proof-of-knowledge manner.

In the remainder of the paper, we describe the new protocol in Section 2, analyze its security and performance in Section 3, and conclude in Section 4.

2 Showing Knowledge of Solovay-Strassen Primality

2.1 Notations

For a positive integer P , let Z_P^* denote the multiplicative group of elements modulo P . For set S , we write $|S|$ for the number of elements in S . For integers a and b , we write $a|b$ if a divides b , (a, b) for the greatest common divisor of a and b , $\left(\frac{a}{b}\right)$ for the Jacobi symbol of a modulo b , and write $\ell(a)$ to denote the size of a , which is the number of the binary bits that a has. For x being a real number, $\lfloor x \rfloor$ denotes the integer part of x (for instance, $\ell(a) = \lfloor \log_2(a) \rfloor + 1$). For $a \in Z_P^*$, we write $Ord_P(a)$ to denote the order of a modulo P .

We shall make use of double exponentiation. Let g be an element of order n , h be an element in Z_n^* , and $x < n$. By double exponentiation of x to the bases h and g we mean

$$g^{(h^x \bmod n)}.$$

2.2 Assumptions and Proof System Setup

We assume that it is computationally infeasible to compute discrete logarithm in the group generated by g . This assumption is reasonable if the group order n is sufficiently large and is non-smooth.

Let Alice be a prover and Bob be a verifier. Alice has constructed an RSA modulus $n = pq$ satisfying that p and q are different odd primes with $\ell(p) = \ell(q)$ or $\ell(p) = \ell(q) \pm 1$.

(We shall later see that our technique can accommodate a suitably bigger size difference if that is desirable.)

She shall then help Bob to setup a multiplicative group of order n . For her part of the setting-up, she only needs to generate a prime P with $n \mid P-1$. This prime can be constructed by testing the primality of $P = \alpha n + 1$ for some even integer α . Once P is found prime, she shall send the numbers n and P to Bob.

Upon receipt of n and P , Bob should test the primality of P , check that n is not a square number, and use trial-division to make sure that n does not have small factors less than 100. Upon passing of these simple tests, he shall fix an element g in Z_P^* using the following procedure. Randomly choose an element β less than P ; test that for every $d \mid \alpha (= (P-1)/n)$, and for $d = n$: $\beta^d \not\equiv 1 \pmod{P}$. Upon finding β satisfying these, g is set to

$$g = \beta^\alpha \pmod{P}.$$

By the prime number theorem, $\alpha (= (P-1)/n)$ is within the scope of $\log_2(P)$. So it will be computationally easy for Bob to completely factor α and therefore use the procedure to find g .

Lemma 1 *Without the knowledge of the factorization of n , the element g which is fixed using the above procedure satisfies $\text{Ord}_P(g) = n$ with an error probability $\frac{\sum_{d \mid n, d \neq n} \phi(d)}{P-1}$.*

Proof Since $g^n \equiv 1 \pmod{P}$, we know $\text{Ord}_P(g) \mid n$. We also know that for any $d \mid P-1$, the cyclic group Z_P^* has exactly $\phi(d)$ elements of order d . So without the knowledge of the factorization of n , there will only be a

$$\frac{\sum_{d \mid n, d \neq n} \phi(d)}{P-1}$$

probability for Bob to pick a g of order less than n . □

This probability is negligible as it is comparable to that of factoring n via trial division.

Bob shall send g to Alice. She shall set

$$A = g^P \pmod{P}, \quad B = g^n \pmod{P}.$$

Alice then sends A, B to Bob. The values g, A, B, n, P will be used as the common input to a protocol for proof that $\log_g(A)$ and $\log_g(B)$ are in fact the two different prime factors of n .

2.3 The Protocol

The protocol Two_Prime_Product specified below demonstrates that the non-square number n is the product of two different odd primes of roughly equal sizes. For clearness, we shall omit the trailing operation mod P in the presentation.

Two_Prime_Product(g, A, B, n, P)

Repeat the following steps k times

1. Bob picks $h \in Z_n^*$ at random, with $\left(\frac{h}{n}\right) = -1$, and sends it to Alice:

2. Alice picks u, v at random, satisfying

$$\ell(u) = \ell((p-1)/2), \quad \ell(v) = \ell((q-1)/2),$$

and sets

$$\begin{aligned} G_U &\leftarrow g^{2u}, & G_V &\leftarrow g^{2v}, \\ A_V &\leftarrow A^{2v}, & B_U &\leftarrow B^{2u}, \\ H_U &\leftarrow B^{(h^u \bmod n)}, & H_V &\leftarrow A^{(h^v \bmod n)}; \end{aligned}$$

Alice sends to Bob: $G_U, G_V, A_V, B_U, H_U, H_V$;

3. Bob picks a challenge $c \in \{0, 1\}$ at random, and sends it to Alice:

4. Alice sends Bob the responses

$$r \leftarrow u + c(p-1)/2, \quad s \leftarrow v + c(q-1)/2;$$

5. Bob verifies (rejects proof if any verification fails):

$$5.1 \quad \ell(r) \leq \lfloor \ell(n)/2 \rfloor + 2, \quad \ell(s) \leq \lfloor \ell(n)/2 \rfloor + 2;$$

$$5.2 \quad g^{2r+1} \equiv \begin{cases} G_U g & c=0 \\ G_U A & c=1 \end{cases}, \quad B^{2s+1} \equiv \begin{cases} B_U B & c=0 \\ B_U & c=1 \end{cases},$$

$$5.3 \quad g^{2s+1} \equiv \begin{cases} G_V g & c=0 \\ G_V B & c=1 \end{cases}, \quad A^{2r+1} \equiv \begin{cases} A_V A & c=0 \\ A_V & c=1 \end{cases},$$

$$5.4 \quad B^{(h^r \bmod n)} \equiv \begin{cases} H_U & c=0 \\ H_U^{\mp 1} & c=1 \end{cases}, \quad A^{(h^s \bmod n)} \equiv \begin{cases} H_V & c=0 \\ H_V^{\mp 1} & c=1 \end{cases};$$

where $H_U^{\pm 1}$ and $H_V^{\mp 1}$ mean that the two exponents take opposite signs.

We should point out that requiring Bob to choose h with the Jacobi symbol -1 is only for conciseness in the protocol specification. Evaluation of the Jacobi symbol can be omitted. Then what is in addition for Bob to check is that in the k iterations, he must demand to see in step 5.4 at least one occasion of H_U, H_V^{-1} , and one occasion of H_U^{-1}, H_V . A simplification can be as follows. Bob begins with choosing h with the Jacobi symbol -1 , and then stops evaluation of the Jacobi symbol once he has seen both of the two occasions. He will have a high probability to see the both occasions within first few iterations, since the probability of keeping on seeing one occasion diminishes exponentially fast. This point will become clear after the proof of Theorem 2.

Although we have required Alice to choose p and q satisfying $\ell(p) = \ell(q)$, or $\ell(p) = \ell(q) \pm 1$, a more relaxed size difference can be accommodated. For each added bit difference, the two size checking inequalities in the protocol step 5.1 should be adjusted accordingly such that 1 is added to the right-hand sides of the two inequalities in 5.1. Later we shall further see that (in the proof of the soundness of the protocol (Section 3.2)) a bigger size difference requires to increase the number of trial division to guard against n containing small factors.

Using a cryptographically secure hash function to generate the challenge bits, the protocol can be modified into parallel version which can save the communication cost.

3 Analyses

We analyze the security and the performance of the protocol. The security consists of the completeness, soundness, and privacy.

3.1 Completeness

Theorem 1 *Suppose Alice has input the correct values (as specified) into the protocol. Then Bob will always accept Alice's proof in each iteration.*

Proof We show that Bob will be satisfied by his verification steps in 5.1 through 5.4.

In 5.1, we note that u has been chosen to satisfy $\ell(u) = \ell((p-1)/2)$. So either $r = u$ or $r = u + (p-1)/2$ will yield (noting $(p-1)/2$ is an integer)

$$\ell(r) \leq \ell((p-1)/2) + 1 = \ell(p) \leq \lfloor \ell(n)/2 \rfloor + 2.$$

(When $\ell(p) = \ell(q)$, the right-hand-side of the inequality can use $\lfloor \ell(n)/2 \rfloor + 1$.)

Analogously, $\ell(s) \leq \lfloor \ell(n)/2 \rfloor + 2$. So the verification in 5.1 will pass.

In the following, we assume the case of challenge $c = 1$. The case $c = 0$ renders the congruences in 5.2 to 5.4 to hold trivially.

In 5.2, with noting $\log_g(A) = p$ and the structures of G_U and r , it is easy to see that the first congruence will hold. Holding of the second congruence is similar, with an attention that $B^p \equiv g^{np} \equiv g^n \equiv 1 \pmod{P}$. So both congruences in 5.2 will hold.

Analogously, both congruences in 5.3 will hold.

To see the congruences in 5.4, observe

$$B^{(h^{(p-1)/2} \bmod n)} \equiv B^{(h^{(p-1)/2} \bmod p)} \equiv B^{\pm 1} \pmod{P}. \quad (1)$$

The first congruence in (1) is due to $\text{Ord}_P(B) = p \mid n$. Then, since p is prime, the second congruence in (1) follows the Euler's criterion. Therefore, the first congruence in 5.4 is

$$\begin{aligned}
B^{(h^r \bmod n)} &\equiv B^{(h^{u+(p-1)/2} \bmod n)} \\
&\equiv (B^{(h^{(p-1)/2} \bmod n)})^{(h^u \bmod n)} \\
&\equiv (B^{(h^{(p-1)/2} \bmod p)})^{(h^u \bmod n)} \\
&\equiv (B \pm 1)^{(h^u \bmod n)} \\
&\equiv (B^{(h^u \bmod n)})^{\pm 1} \\
&\equiv H_P^{\pm 1} \pmod{P}.
\end{aligned}$$

Indeed, Bob will accept this congruence.

Analogously, the second congruence in 5.4 will hold.

Finally, we note that (1) actually evaluates the Jacobi symbol $\left(\frac{h}{p}\right)$. Since Bob has chosen h satisfying

$$\left(\frac{h}{n}\right) = \left(\frac{h}{p}\right) \left(\frac{h}{q}\right) = -1,$$

the exponents of $H_P^{\pm 1}$ and $H_Q^{\mp 1}$ must take opposite signs. □

3.2 Soundness

We begin with emphasizing that all the numbers and variables to appear in this section are non-negative integers. In particular, $\log_g(A)$ and $\log_g(B)$ denote the least positive integers x and y less than $\text{Ord}_P(g)$ satisfying $A \equiv g^x$ and $B \equiv g^y$.

Lemma 2 *When Bob accepts a proof on the values input to Two_Prime_Product, he accepts*

$$i) \log_g(A) \leq 2^{\lfloor \ell(n)/2 \rfloor + 3}, \quad \log_g(B) \leq 2^{\lfloor \ell(n)/2 \rfloor + 3},$$

$$ii) \log_g(A) = a \text{Ord}_P(B), \quad \log_g(B) = b \text{Ord}_P(A) \text{ where } a \leq 64 \text{ and } b \leq 64,$$

with probability $> 1 - 1/2^k$ where k is the number of iterations in the protocol.

Proof

For clarity, we denote $x = \text{Ord}_P(B)$, $y = \text{Ord}_P(A)$. Since A and B are generated from g of order n (recall Lemma 1, $\text{Ord}_P(g) = n$ holds with probability $1 - \frac{\sum_{d|n, d \neq n} \phi(d)}{P-1}$), we have

$$x | n, \quad y | n. \tag{2}$$

The verification of the first congruence in 5.2 forms a standard proof of knowledge on that Alice knows $\log_g(A)$, and the second congruence further shows

$$B^{\log_g(A)} \equiv 1 \pmod{P}. \tag{3}$$

Analogously, 5.3 forms the verification of proof on Alice knowing $\log_g(B)$ and

$$A^{\log_g(B)} \equiv 1 \pmod{P}. \tag{4}$$

The congruences (3) and (4) imply

$$x \mid \log_g(A), \quad y \mid \log_g(B),$$

So we can write

$$\log_g(A) = ax, \quad \log_g(B) = by \quad (5)$$

for some integers a and b .

In protocol step 5.1. Bob has checked that for both challenge cases, the responses r and s satisfy

$$\ell(r) \leq \lfloor \ell(n)/2 \rfloor + 2, \quad \ell(s) \leq \lfloor \ell(n)/2 \rfloor + 2.$$

Since $\ell(\log_g(A)) \leq \ell(2r)$ and $\ell(\log_g(B)) \leq \ell(2s)$, so

$$\ell(\log_g(A)) \leq \lfloor \ell(n)/2 \rfloor + 3, \quad \ell(\log_g(B)) \leq \lfloor \ell(n)/2 \rfloor + 3,$$

or (we have proved the claim (i))

$$\log_g(A) \leq 2^{\lfloor \ell(n)/2 \rfloor + 3}, \quad \log_g(B) \leq 2^{\lfloor \ell(n)/2 \rfloor + 3}. \quad (6)$$

Combining this with (5), we have

$$axby = \log_g(A) \log_g(B) \leq 2^{\ell(n)+6},$$

or

$$abxy \leq 64n. \quad (7)$$

Applying the following number theoretic fact (with t in a cyclic group)

$$\text{Ord}(t^k) = \text{Ord}(t) / (\text{Ord}(t), k),$$

with noticing $x = \text{Ord}_P(B)$, $B = g^{\log_g(B)}$, $\text{Ord}_P(g) = n$, $\log_g(B) = by$, we can derive

$$x = n / (n, by).$$

Noticing (2), y is a factor of n , so $(n, by) = b'y$ for some $b' \leq b$. Thus we have reached

$$b'xy = n. \quad (8)$$

Placing this result and $b' \leq b$ into (7), we derive

$$ab \leq 64b' \leq 64b.$$

So $a \leq 64$. By symmetry we can also show $b \leq 64$. These plus (5) form claim (ii). (The inequalities in (6) have proved claim (i).) Clearly, if any of the claims is not true, then in each iteration Alice will have at most $1/2$ probability to correctly answer Bob's random challenge. Therefore the claims hold with the probability $1 - 1/2^k$ after k iterations of acceptance. \square

Theorem 2 Let n be a non-square integer, be free of factors up to 100, and be accepted by a proof running TwoPrime_Product. Then with probability $> 1 - 1/2^k$

i) $\log_g(A)$ and $\log_g(B)$ are different odd primes,

ii) $n = \log_g(A) \log_g(B)$,

where k is the number of iterations in TwoPrimeProduct.

Proof

We shall use the following notations defined in the proof of Lemma 2: $x = \text{Ord}_P(B)$, $y = \text{Ord}_P(A)$, $ax = \log_g(A)$, $ay = \log_g(B)$.

Bob's verification of the two congruences in 5.4 (combined with the verification in 5.2 and 5.3) show him that (when $c = 1$)

$$B^{(h^{(ax-1)/2} \bmod n)} \equiv B^{\pm 1} \pmod{P}, \quad (9)$$

and

$$A^{(h^{(by-1)/2} \bmod n)} \equiv A^{\mp 1} \pmod{P}. \quad (10)$$

for h relatively prime to n with the Jacobi symbol -1 . Noticing (4), $x | n$ and $y | n$, the congruences (9) and (10) imply that for such h ,

$$h^{(ax-1)/2} \equiv \pm 1 \pmod{x} \quad (11)$$

and

$$h^{(by-1)/2} \equiv \mp 1 \pmod{y}.$$

These results allow us to apply a variation of Solovay-Strassen primality test technique [9] to prove the theorem. Define H_x as the following set

$$H_x = \{ h \in Z_x^* \mid (h, x) = 1 \ \& \ h^{(ax-1)/2} \equiv \pm 1 \pmod{x} \ \& \ a \text{ is an integer} \}.$$

Obviously, for any a , H_x is a subgroup of Z_x^* (at least it contains 1), and we know $H_x = Z_x^*$ from Euler's criterion when $a = 1$ and x is prime. We shall prove on the other hand that, if $a \neq 1$, or if x is composite, then H_x will be a proper subgroup of Z_x^* . This renders $|H_x| \leq |Z_x^*|/2$, and thus randomly picking $h \in Z_n^*$ as Bob did in each iteration in the protocol, the probability for $h \bmod x$ to fall in H_x cannot exceed $1/2$.

There are several cases need to be reasoned. We shall reason all the cases of x under $a \neq 1$. The cases under $a = 1$ can be analogously reasoned by following the cases we reason below.

Suppose $a \neq 1$. There are three sub-cases for x .

Case 1: x is prime. Then (11) implies

$$h^{ax-1} \equiv 1 \pmod{x},$$

for h not divisible by x . Since Z_x^* is of order $x-1 < ax-1$ ($a > 1$), we have $x-1 | ax-1$. So there exists an integer d such that $d(x-1) = ax-1$, or $x = 1 + (a-1)/(d-a)$. In

Lemma 2 we have proved $a \leq 64$. So $x < 64$. From the hypothesis of the theorem such x cannot be a factor n . So when $a \neq 1$, x must be composite.

Case 2: $x = r^d$ with r being prime and $d > 1$. In the group $Z_{r^d}^*$ there exists an element e of the full order $(r-1)r^{d-1}$. This e cannot be in H_x since otherwise (11) will imply

$$e^{ar^d-1} \equiv 1 \pmod{r^d},$$

yielding

$$(r-1)r^{d-1} \mid ar^d - 1.$$

So there exists λ satisfying

$$ar^d - \lambda(r-1)r^{d-1} = 1.$$

This means $(r-1)r^{d-1}$ is relatively prime to r^d , impossible with $d > 1$. Thus, for this case H_x must be a proper subset of Z_x^* .

Case 3: The only remaining possibility for x being composite renders a non-trivial factorization $x = \xi\eta$ with $(\xi, \eta) = 1$. We claim that, either H_x is a proper subset of Z_x^* (and we have done), or $H_x = Z_x^*$ will only contain elements satisfying

$$h^{(ax-1)/2} \equiv 1 \pmod{x}. \quad (12)$$

Suppose $H_x = Z_x^*$ while (12) is not true. There exists an element $e \in H_x = Z_x^*$ with $e^{(ax-1)/2} \equiv -1 \pmod{x}$. Since ξ and η are relatively prime, by the Chinese remainder theorem, the system $z \equiv 1 \pmod{\xi}$, $z \equiv e \pmod{\eta}$ has a solution $f \in Z_x^*$. Obviously,

$$f^{(ax-1)/2} \equiv 1 \pmod{\xi}, \quad f^{(ax-1)/2} \equiv -1 \pmod{\eta},$$

yielding

$$f^{(ax-1)/2} \not\equiv \pm 1 \pmod{x}.$$

So $f \in Z_x^* \setminus H_x$, contradiction to $H_x = Z_x^*$.

So now we have to consider $H_x = Z_x^*$ with holding of the condition (12). That will cause Bob to see in a proof that, when the challenge is $c = 1$, the right-hand side of the first congruence in 5.4 will always take the positive exponent, and hence, that of the second congruence will always take the negative (-1) exponent. This means that Case 3 will not apply to both x and y . So when x is in Case 3, y can only be in the form of r^d with r prime and $d \geq 1$. We first consider $b \neq 1$ ($by = \log_y(B)$, defined in (5) in Lemma 2). Then (by symmetry) either y will be a prime factor of n satisfying $y < 64$ (Case 1) which has been excluded by the hypothesis of the theorem, or H_y (definition of H_y follows that of H_x by symmetry) will be a proper subset of Z_y^* (Case 2) which will cause Bob to reject each iteration with at least $1/2$ probability. Finally we consider $b = 1$. Then there are only exactly half of elements in Z_y^* ($= Z_{r^d}^*$) which are quadratic non-residue modulo y satisfying

$$h^{(y-1)/2} \equiv -1 \pmod{y}, \quad (13)$$

So in each iteration the probability for Bob to pick an h with $h \bmod y$ satisfying (13) is only $1/2$.

To this end we have proved all the cases of x under $a \neq 1$. The remaining cases under $a = 1$ can be analogously proved. The proofs are very similar to those above, except that Case 1 is no longer needed.

By symmetry, we know $b = 1$ and y is prime. Since $a = b = 1$ and in (8) $b' \leq b = 1$, we have finally proved

$$\log_g(A) \log_g(B) = xy = n.$$

The two primes are different since n is not a square number.

We have fully proved the theorem. □

Discussions

We have seen in the proofs in Lemma 2 and in Theorem 2 that the size difference between the two prime factors of n can be set to a desirable scope. The bigger the difference of their sizes, the more work (e.g., trial division) is needed to guard against n containing trivially small factors as in Case 1. In the specified protocol, we have allowed for the size difference to be 1 bit.

In Section 2.2 we mentioned that Bob can stop evaluation of the Jacobi symbols once he has seen both occasions of H_U , H_V^{-1} and H_U^{-1} , H_V in protocol step 5.4. The reason for this now becomes clear: an appearance of a negative exponent (one show enough) shows the impossibility of Case 3 for the respective factor of n .

3.3 Privacy

Theorem 3 *Assume the computational infeasibility of computing discrete logarithms to the base g , of factoring n , and of determining $\left(\frac{h}{x}\right)$ for x being a factor of n with given g^x . Then on inputting $n = pq$ with p and q being odd primes, the protocol Two_Prime_Product is perfect computational zero-knowledge.*

Proof

First, we note that in protocol step 2, Alice picks two random numbers u and v with $\ell(u) = \ell((p-1)/2)$ and $\ell(v) = \ell((q-1)/2)$. Since p and q are odd, it is clear that for the challenge case $c = 1$, the responses do not disclose any information about p and q .

Now on the input tuple (g, A, B, n, P) , each iteration in a protocol run can be simulated perfectly by a simulator in polynomial time. The simulator picks the following values at uniformly random (which means it picks h and c exactly as Bob does, and picks r and s exactly as Alice picks u and v):

$$h \in Z_n^* \text{ satisfying } \left(\frac{h}{n}\right) = -1;$$

$c \in \{0, 1\}$; if $c = 1$ then further picks $d \in \{1, -1\}$;

r, s with $\ell(r) = \ell(s)$ or $\ell(r) = \ell(s) \pm 1$ satisfying that $\ell(r) \leq \lfloor \ell(n)/2 \rfloor + 2$,
 $\ell(s) \leq \lfloor \ell(n)/2 \rfloor + 2$.

Using these random values, he then computes the following values to construct a simulated view.

For $c = 0$:

$$G_U \leftarrow g^{2r} \bmod P, \quad G_V \leftarrow g^{2s} \bmod P,$$

$$H_U \leftarrow B^{(h^r \bmod n)} \bmod P, \quad H_V \leftarrow A^{(h^s \bmod n)} \bmod P;$$

For $c = 1$:

$$G_U \leftarrow g^{2r+1}/A \bmod P, \quad G_V \leftarrow g^{2s+1}/B \bmod P,$$

$$H_U \leftarrow (B^{(h^r \bmod n)})^d \bmod P, \quad H_V \leftarrow (A^{(h^s \bmod n)})^{-d} \bmod P;$$

Under the hypotheses of the theorem, the values G_U, G_V, H_U, H_V will have exactly the same distribution as those in a true proof interaction. Therefore, the simulation is perfect. Consequently, Bob gains no additional information about p and q other than from n . (We point out that since g is a quadratic residue, so for both cases of c , all values computed above are quadratic residues modulo P .) \square

3.4 Performance

Compared with the previous approaches of applying proofs of Blum, square-free, and showing factor sizes, the new protocol saved the need of agreeing mutually trusted challenges, and saved the additional cost on showing factor sizes.

Since it is generally accepted that Solovay-Strassen primality test (SSPT) on a non-secret number is very efficient (though Miller-Rabin will be more efficient), we shall make a precise performance comparison between our protocol and SSPT. We consider k iterations in our protocol and in an instance of performing SSPT.

First, we shall not consider the cost for Alice to setup n since it is irrelevant to the cost of the proof protocol. Thus, the system setting-up involves one primality test on P (by both Alice and Bob), and fixing the element g (by Bob). Note that P is non-smooth because $P - 1$ contains n as a factor. So almost all elements in Z_P^* satisfy the requirements of g . Thus, the cost of this part is roughly that of the primality test on P . The test can use Solovay-Strassen. By the prime number theorem, the size of P can be bounded above by $\log_2(n) + \log_2(\log_2(n))$.

Recall that we have pointed out that in the real use of our protocol, Bob will only need to evaluate very few Jacobi symbols (reason given in the discussion at the end of Section 3.2). SSPT needs to evaluate $2k$ Jacobi symbols. Nevertheless, evaluation of the Jacobi symbols are very efficient compared with the modulo exponentiations; so we shall omit this part of the comparison.

Next, checking of the sizes of the responses in our protocol step 5.1 takes trivial computation. So we shall omit counting that cost too.

Now observe that in our protocol, in each iteration, Alice and Bob computes the same number of modulo exponentiations: six exponentiations modulo P , and two of them modulo n . Considering $\ell(P) \approx \ell(n) \approx 2\ell(p)$, together these convert to 16 exponentiations modulo p . This is the real computational cost of our protocol.

In each iteration of SSPT testing the primality of opened p and q , two exponentiations modulo p are required. Thus, in comparison of the numbers of modulo exponentiations on moduli of the same size, the performance of our protocol counts for that of running eight instances of SSPT. Further adding the primality test on P by both parties with $\ell(P) \approx 2\ell(p)$, we conclude the following theorem.

Theorem 4 *The whole computational cost for demonstrating n being the product of two primes of roughly equal sizes using protocol Two_Prime_Product is at the same level of performing 12 primality tests using SSPT on non-secret numbers of size equal to $\ell(n)/2$, or equivalently, 6 such tests on non-secret numbers of size $\ell(n)$. \square*

In terms of the bit operations, probabilistic testing of primality of P counts for $O(k \log_2(P))$ operations. Considering $\log_2(P) \approx \log_2 n + \log_2 \log_2 n$, the computational cost of the protocol can be measured by $O(k(\log_2 n + \log_2 \log_2 n))$ bit operations.

We should point out that in this estimate we have added the computations of Alice and Bob. If a standard parallelization proof technique is used (using a cryptographically secure hash function to generate challenges), then a proof can be viewed as a certificate for the structure of n . In that case, the cost of a verification will only count half of that given by Theorem 4.

To the author's knowledge, this result is better than those of all known previous protocols that can prove the structure of n being the product of exactly two primes with structure being congruent to 3 modulo 4, yet without showing their sizes. We therefore believe that the cost of the proposed protocol reaches a new low.

4 Conclusion

We have constructed an efficient proof of knowledge protocol for demonstrating that an integer is the product of two prime factors of roughly equal sizes. The proposed protocol is the first of its kind to prove such a structure with an efficiency comparable to that of

a Monte-Carlo method for primality test on non-secret numbers of comparable sizes. It is also the first of its kind to prove such a structure without limiting to Blum integers.

An open problem is to further demonstrate efficiently the structure of $p \pm 1$ in $n = pq$. Such a demonstration will show further strengths of n against a number of known factoring algorithms.

Acknowledgments

It is a pleasure to thank my colleagues Simon Crouch and Nigel Smart for interesting discussions about the subject.

References

- [1] Berger, R., S. Kannan and R. Peralta. A framework for the study of cryptographic protocols. *Advances in Cryptology — Proceedings of CRYPTO 85* (H.C. Williams ed.), Lecture Notes in Computer Science, Springer-Verlag 218 (1986), pp. 87-103.
- [2] Blum, M. Coin flipping by telephone: a protocol for solving impossible problems. *Proceedings of 24th IEEE Computer Conference (Comp Con)*, pp. 133-137. 1982.
- [3] Boyar, J., K. Friedl and C. Lund. Practical zero-knowledge proofs: Giving hints and using deficiencies. *Advances in Cryptology — Proceedings of EUROCRYPT 89* (J.-J. Quisquater and J. Vandewalle, eds.), Lecture Notes in Computer Science, Springer-Verlag 434 (1990) pp. 155-172.
- [4] Galil, Z., S. Haber and M. Yung. A private interactive test of a boolean predicate and minimum-knowledge public-key cryptosystems. *26th FOCS*, 1985. pp. 360-371.
- [5] Gennaro, R., D. Miccianicio and T. Rabin. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products. In *5th ACM Conference on Computer and Communications Security*, October 1998.
- [6] Van de Graaf, J. and R. Peralta. A simple and secure way to show the validity of your public key. *Advances in Cryptology — Proceedings of CRYPTO 87* (E. Pomerance, ed.), Lecture Notes in Computer Science, Springer-Verlag 293 (1988) pp. 128-134.
- [7] ISO/IEC 9798-3. "Information technology - Security techniques - Entity authentication mechanisms - Part 3: Entity authentication using a public-key algorithm", International Organization for Standardization, Geneva, Switzerland, 1993 (first edition).

- [8] Micali, S. Fair public key cryptosystems. *Advances in Cryptology — Proceedings of CRYPTO 92* (E.F. Brickell, ed.) Lecture Notes in Computer Science Springer-Verlag 740 (1993) pp. 113-138.
- [9] Solovay, R. and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal of Computing*, vol. 6, no. 1, March 1977. pp. 84-85.

CLAIMS

(30990020)

1. A method for proving to a verifier that an integer n , supplied to the verifier by a prover, is an authentic RSA modulus, by proving that n is the product of exactly two different odd prime numbers of similar sizes, the method not being limited to n being a Blum integer and comprising the prover showing the verifier Solovay-Strassen primality evidence in a zero-knowledge proof.
2. A method according to claim 1, comprising an iterative process of challenge-response interaction between the prover and the verifier, wherein, a challenge is chosen at random solely by the verifier, and, for a number of iterations k , the number of bit operations needed to prove that n is an authentic RSA modulus is $O(k(\log_2(n) + \log_2(\log_2(n))))$ with a probability of error being $(\frac{1}{2})^k$.